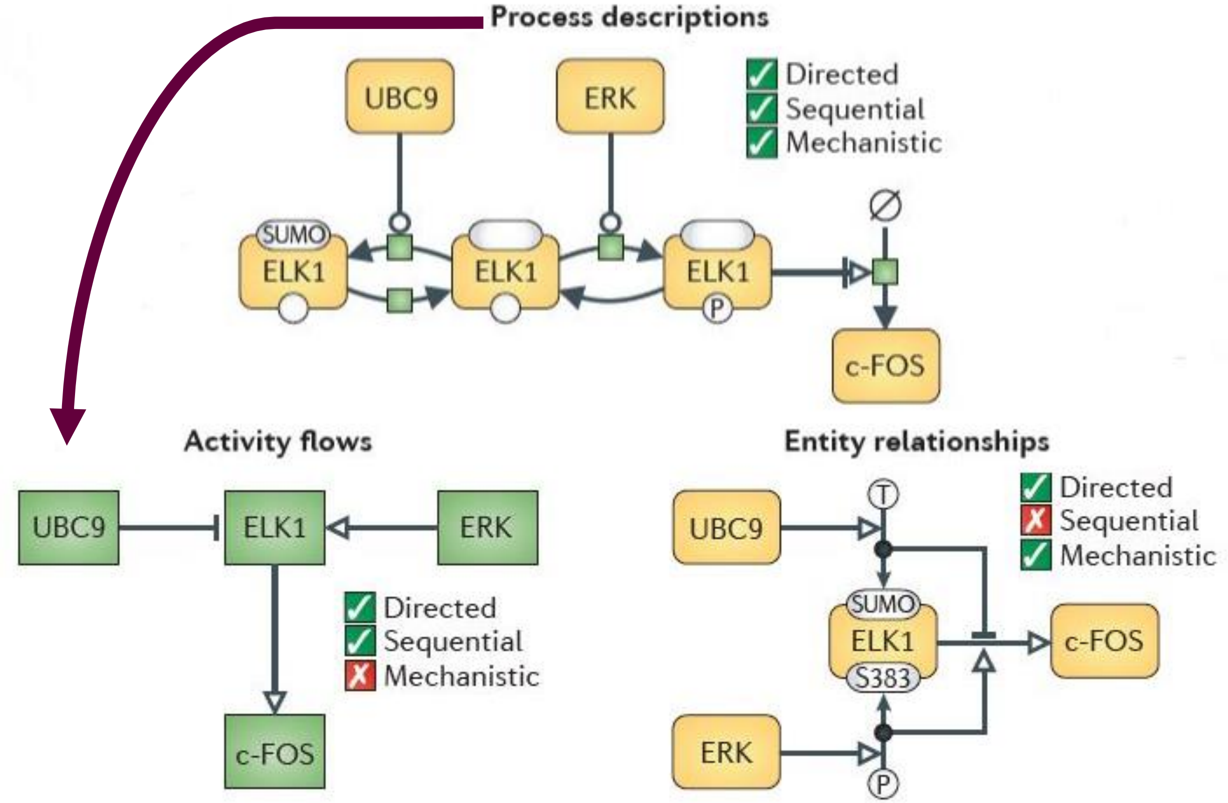
Hands on CaSQ

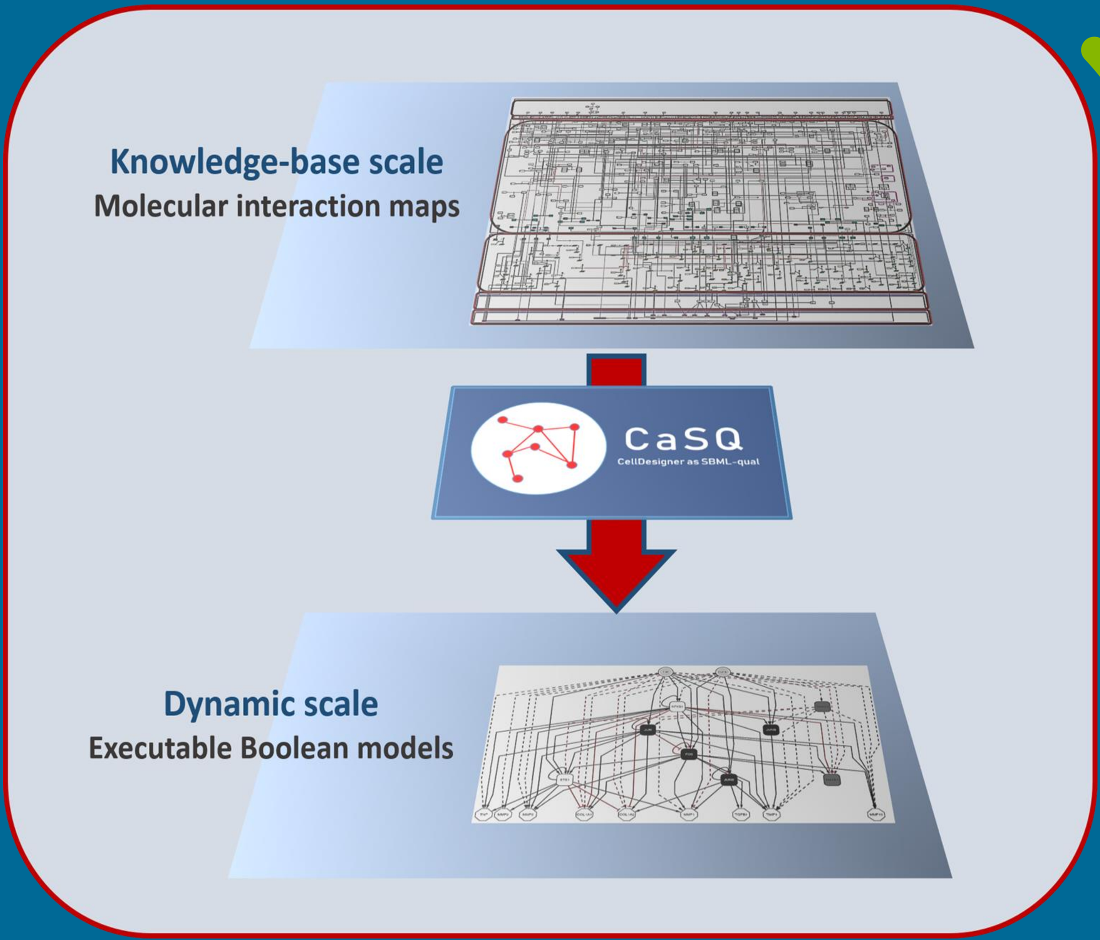
From PD to AF + logical formulae



+ mathematical description

(Le Novere, 2015), (Mazein et al., 2018), (Vogt et al., 2013)

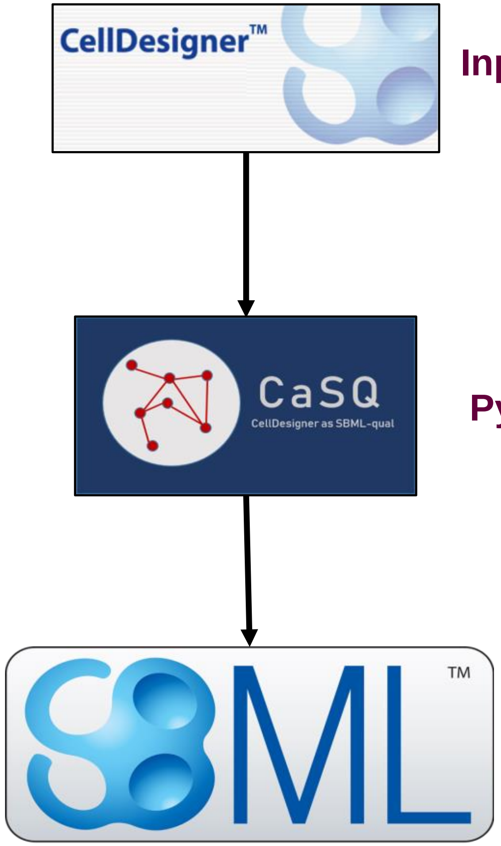
Executable Boolean models from molecular interaction maps



Knowledge-base scale
Molecular interaction maps

CaSQ
CellDesigner as SBML-qual

Dynamic scale
Executable Boolean models

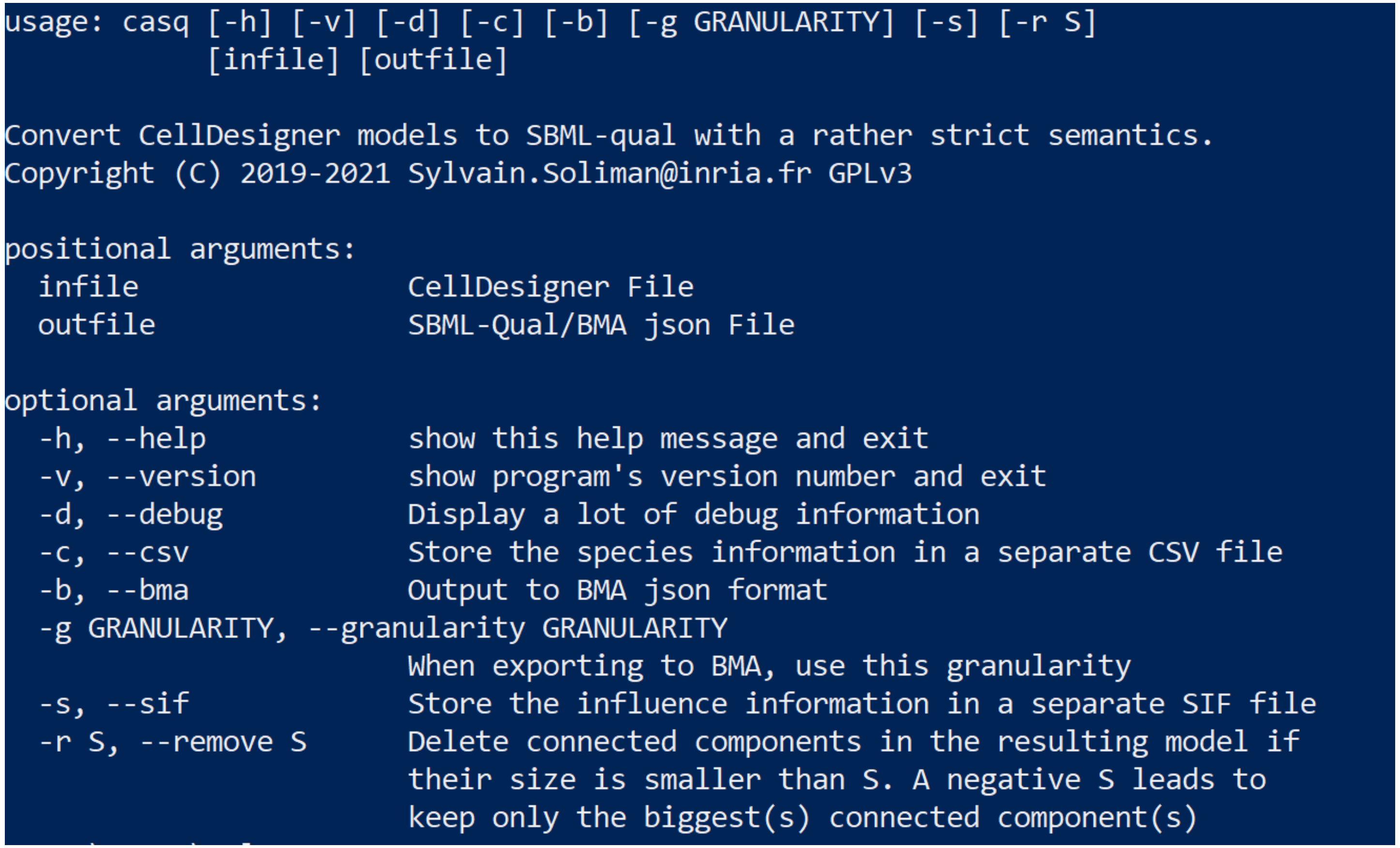


Input: CellDesigner XML file

Python package

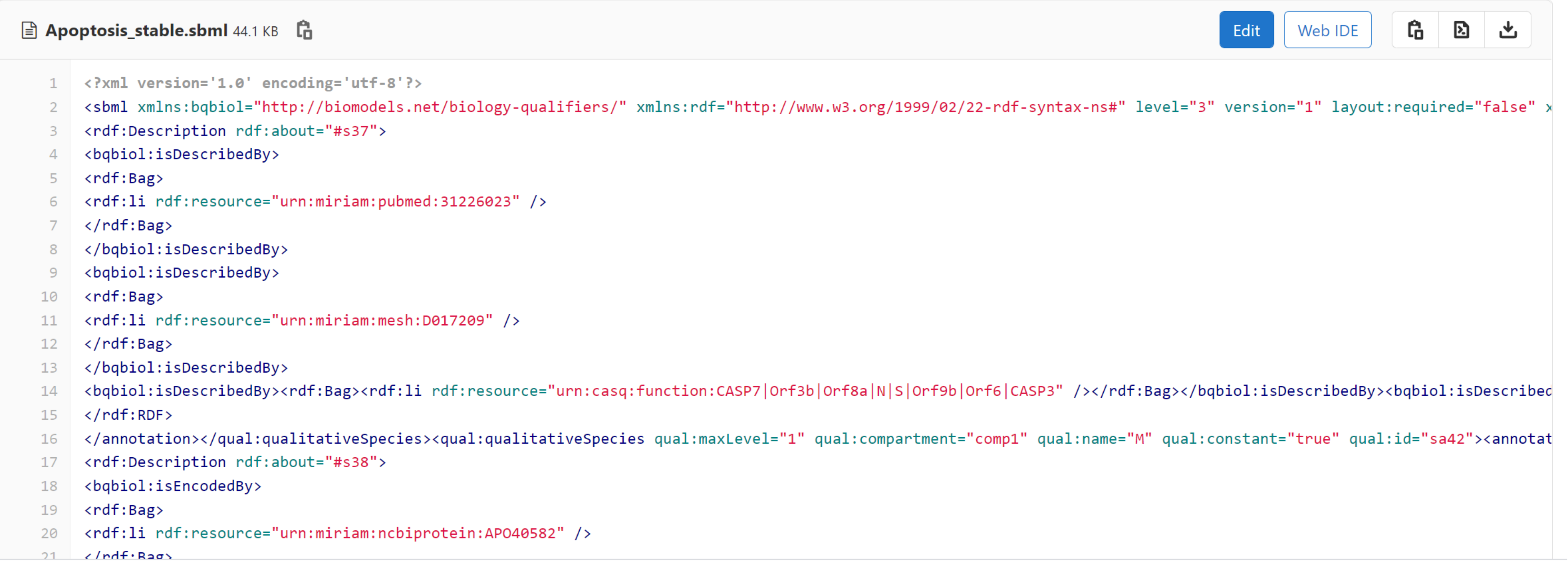
Output: SBML-qual file
Systems Biology Markup Language

CaSQ – a Python package

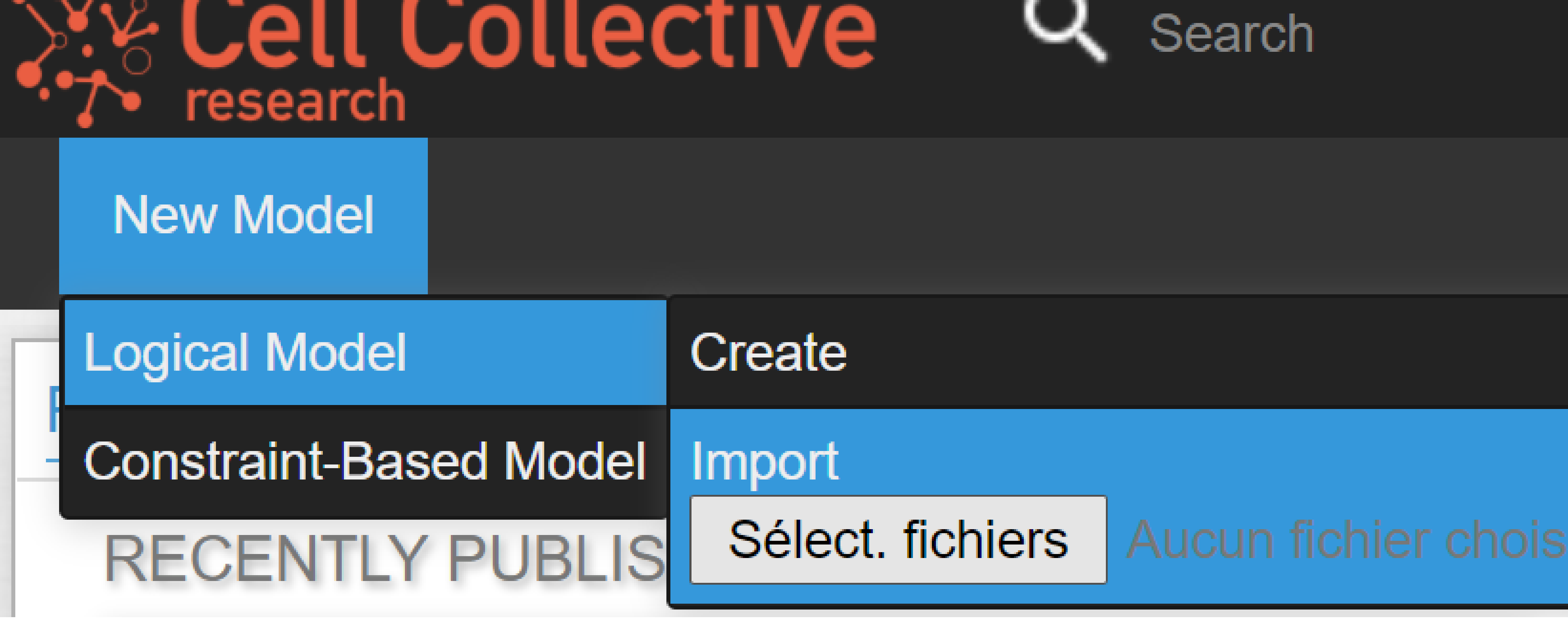


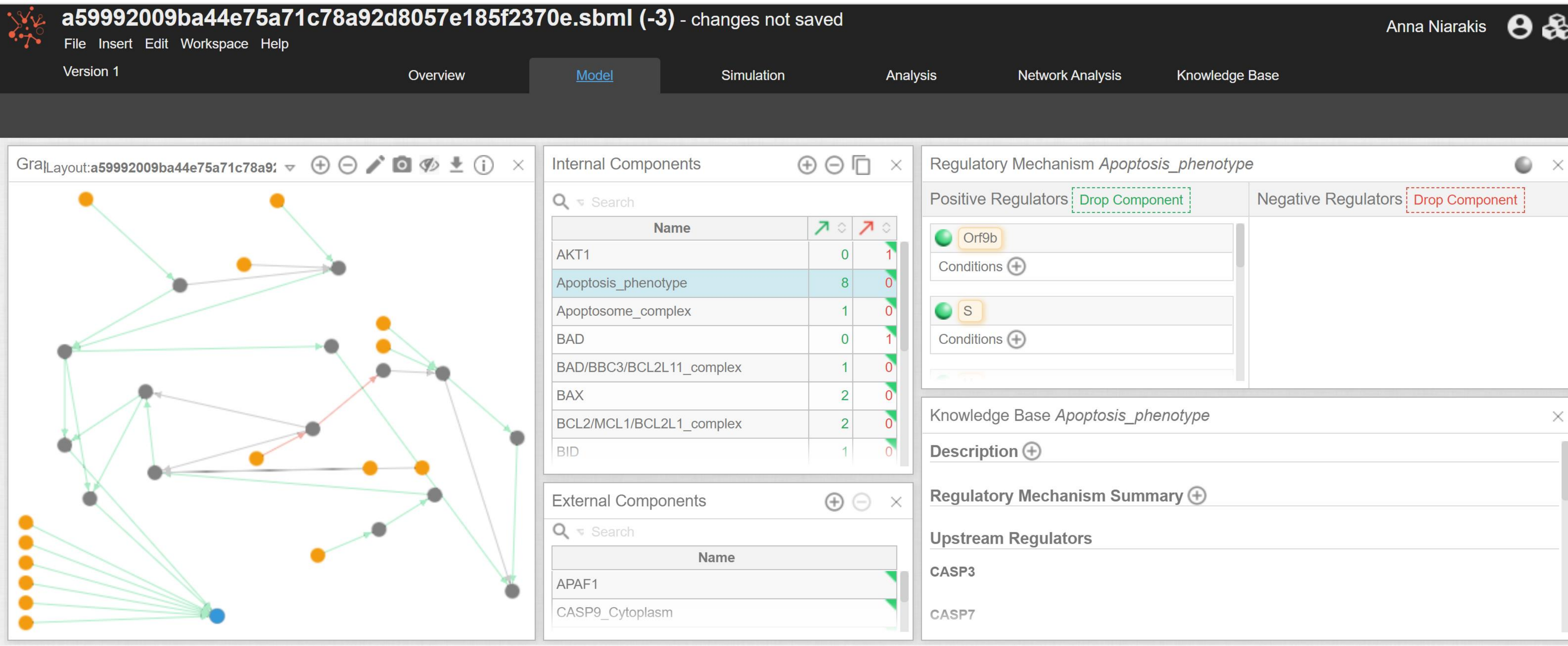
```
usage: casq [-h] [-v] [-d] [-c] [-b] [-g GRANULARITY] [-s] [-r S]
            [infile] [outfile]

Convert CellDesigner models to SBML-qual with a rather strict semantics.
Copyright (C) 2019-2021 Sylvain.Soliman@inria.fr GPLv3

positional arguments:
  infile                CellDesigner File
  outfile               SBML-Qual/BMA json File

optional arguments:
  -h, --help            show this help message and exit
  -v, --version         show program's version number and exit
  -d, --debug           Display a lot of debug information
  -c, --csv             Store the species information in a separate CSV file
  -b, --bma             Output to BMA json format
  -g GRANULARITY, --granularity GRANULARITY
                        When exporting to BMA, use this granularity
  -s, --sif             Store the influence information in a separate SIF file
  -r S, --remove S      Delete connected components in the resulting model if
                        their size is smaller than S. A negative S leads to
                        keep only the biggest(s) connected component(s)
```

SBML qual



```
<?xml version='1.0' encoding='utf-8'?>
<sbml xmlns:bqbiol="http://biomodels.net/biology-qualifiers/" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" level="3" version="1" layout:required="false" x
<rdf:Description rdf:about="#s37">
<bqbiol:isDescribedBy>
<rdf:Bag>
<rdf:li rdf:resource="urn:miriam:pubmed:31226023" />
</rdf:Bag>
</bqbiol:isDescribedBy>
<bqbiol:isDescribedBy>
<rdf:Bag>
<rdf:li rdf:resource="urn:miriam:mesh:D017209" />
</rdf:Bag>
</bqbiol:isDescribedBy>
<bqbiol:isDescribedBy><rdf:Bag><rdf:li rdf:resource="urn:casq:function:CASP7|Orf3b|Orf8a|N|S|Orf9b|Orf6|CASP3" /></rdf:Bag></bqbiol:isDescribedBy><bqbiol:isDescribed
</rdf:RDF>
</annotation></qual:qualitativeSpecies><qual:qualitativeSpecies qual:maxLevel="1" qual:compartment="comp1" qual:name="M" qual:constant="true" qual:id="sa42"><annotat
<rdf:Description rdf:about="#s38">
<bqbiol:isEncodedBy>
<rdf:Bag>
<rdf:li rdf:resource="urn:miriam:ncbiprotein:APO40582" />
</rdf:Bag>
```


Cell Collective
research
Search
New Model
Logical Model
Constraint-Based Model
Create
Import
Sélect. fichiers
Aucun fichier chois
RECENTLY PUBLIS



Internal Components

Name	↗	↗
AKT1	0	1
Apoptosis_phenotype	8	0
Apoptosome_complex	1	0
BAD	0	1
BAD/BBC3/BCL2L11_complex	1	0
BAX	2	0
BCL2/MCL1/BCL2L1_complex	2	0
BID	1	0

External Components

Name
APAF1
CASP9_Cytoplasm

Regulatory Mechanism *Apoptosis_phenotype*

Positive Regulators

- Orf9b — Conditions ⊕
- S — Conditions ⊕

Negative Regulators

Knowledge Base *Apoptosis_phenotype*

Description ⊕

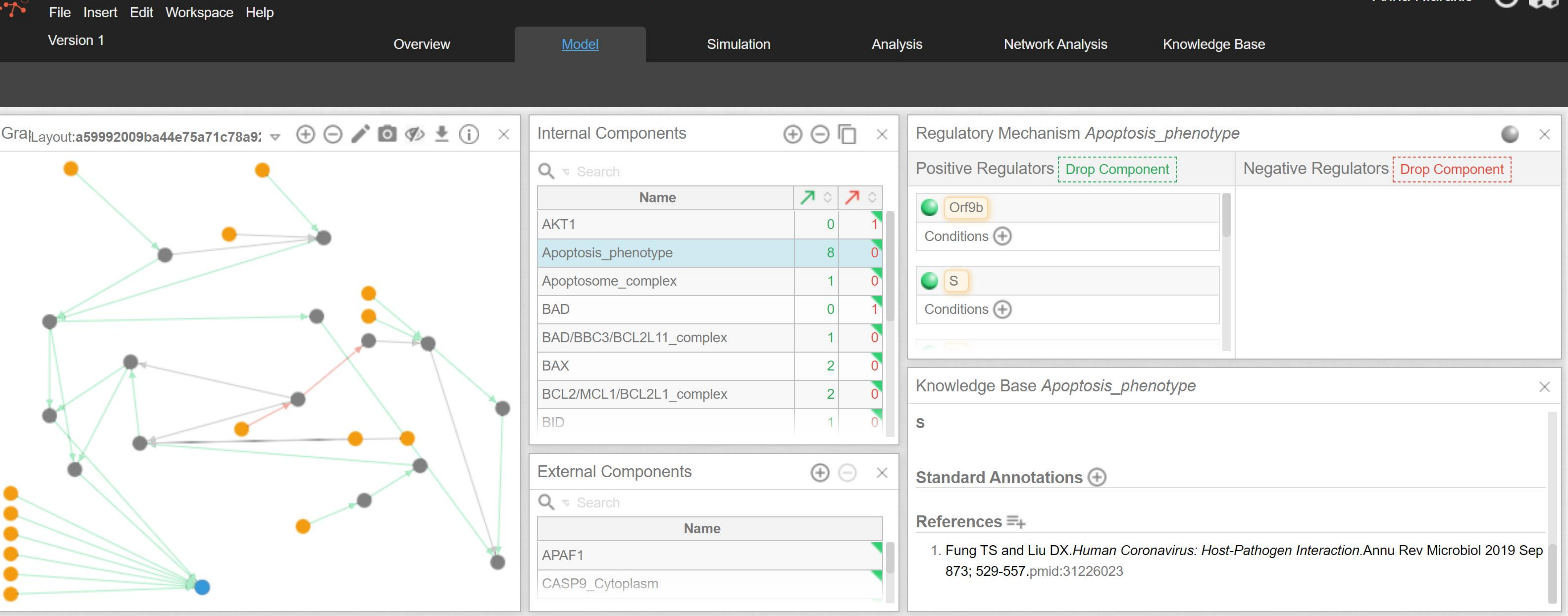
Regulatory Mechanism Summary ⊕

Upstream Regulators

CASP3

CASP7

Layout



Internal Components

Name	↗	↗
AKT1	0	1
Apoptosis_phenotype	8	0
Apoptosome_complex	1	0
BAD	0	1
BAD/BBC3/BCL2L11_complex	1	0
BAX	2	0
BCL2/MCL1/BCL2L1_complex	2	0
BID	1	0

External Components

Name
APAF1
CASP9_Cytoplasm

Regulatory Mechanism *Apoptosis_phenotype*

Positive Regulators: Orf9b, S

Negative Regulators

Knowledge Base *Apoptosis_phenotype*

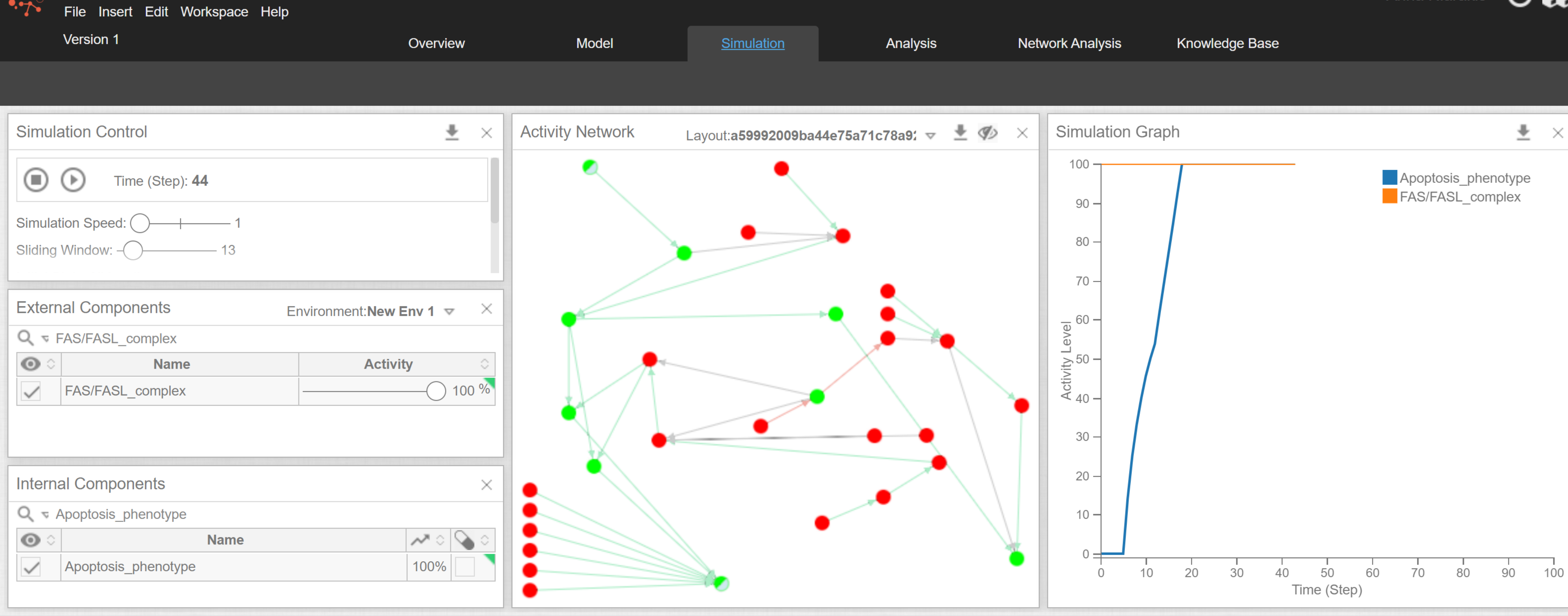
S

Standard Annotations

References

1. Fung TS and Liu DX.*Human Coronavirus: Host-Pathogen Interaction.*Annu Rev Microbiol 2019 Sep 873; 529-557.pmid:31226023

References



Simulation Control

Time (Step): **44**

Simulation Speed: 1

Sliding Window: 13

External Components

Environment:**New Env 1**

FAS/FASL_complex

	Name	Activity
✓	FAS/FASL_complex	100 %

Internal Components

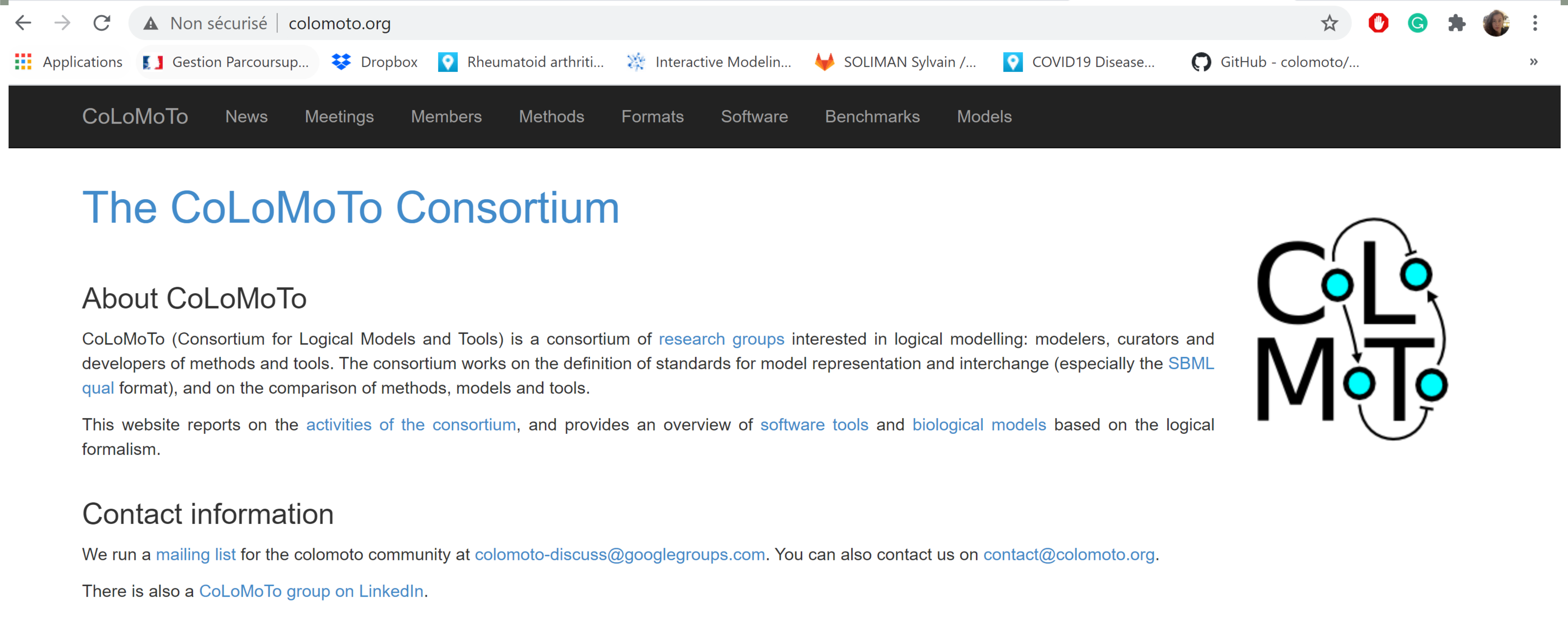
Apoptosis_phenotype

	Name		
✓	Apoptosis_phenotype	100%	□

Layout:**a59992009ba44e75a71c78a9:**

Real time Simulations

The CoLoMoTo consortium



The CoLoMoTo Consortium

About CoLoMoTo

CoLoMoTo (Consortium for Logical Models and Tools) is a consortium of research groups interested in logical modelling: modelers, curators and developers of methods and tools. The consortium works on the definition of standards for model representation and interchange (especially the SBML qual format), and on the comparison of methods, models and tools.

This website reports on the activities of the consortium, and provides an overview of software tools and biological models based on the logical formalism.

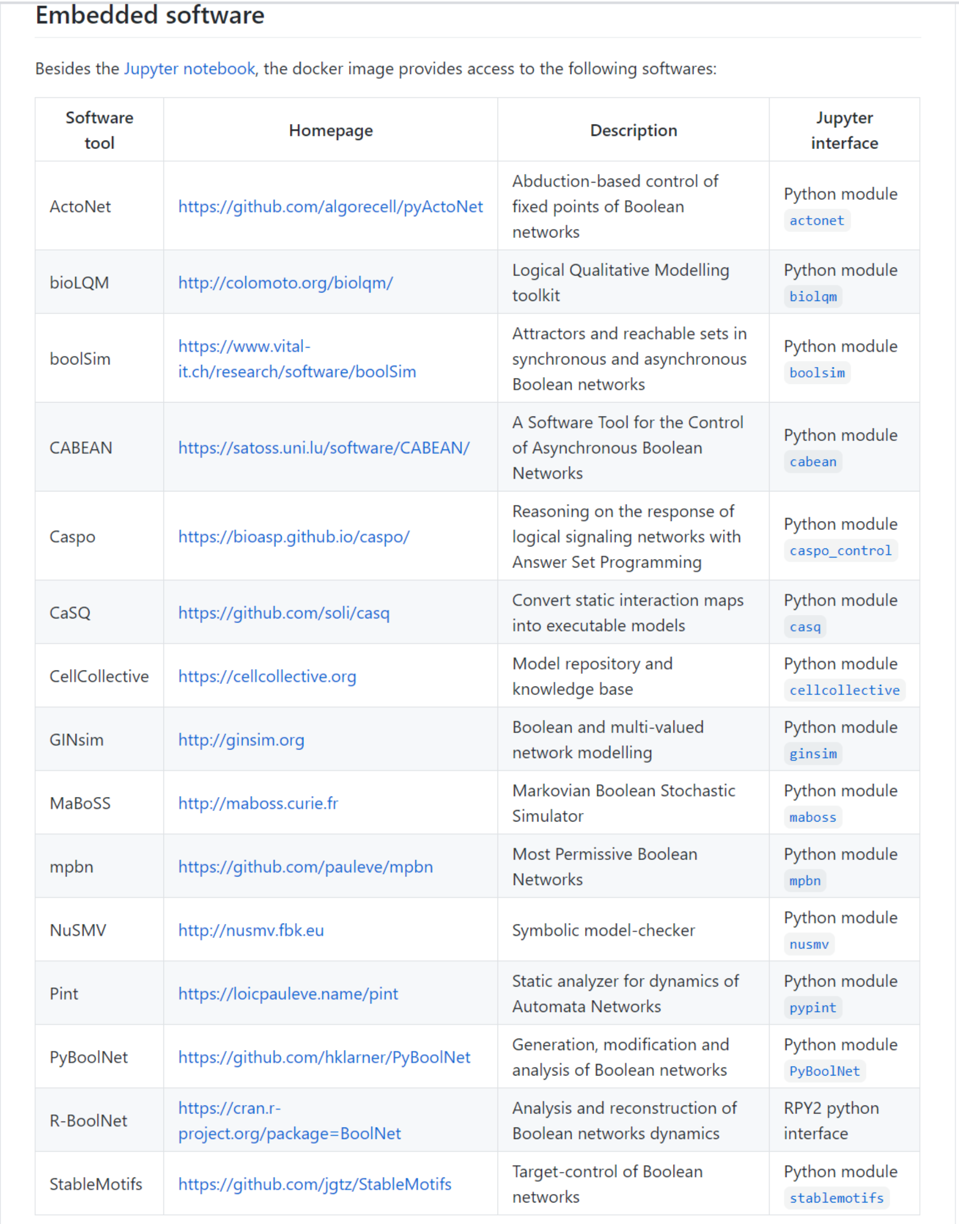
Contact information

We run a mailing list for the colomoto community at colomoto-discuss@googlegroups.com. You can also contact us on contact@colomoto.org.

There is also a CoLoMoTo group on LinkedIn.

The CoLoMoTo notebook

- Visit it!
- https://github.com/colomoto/colomoto-docker

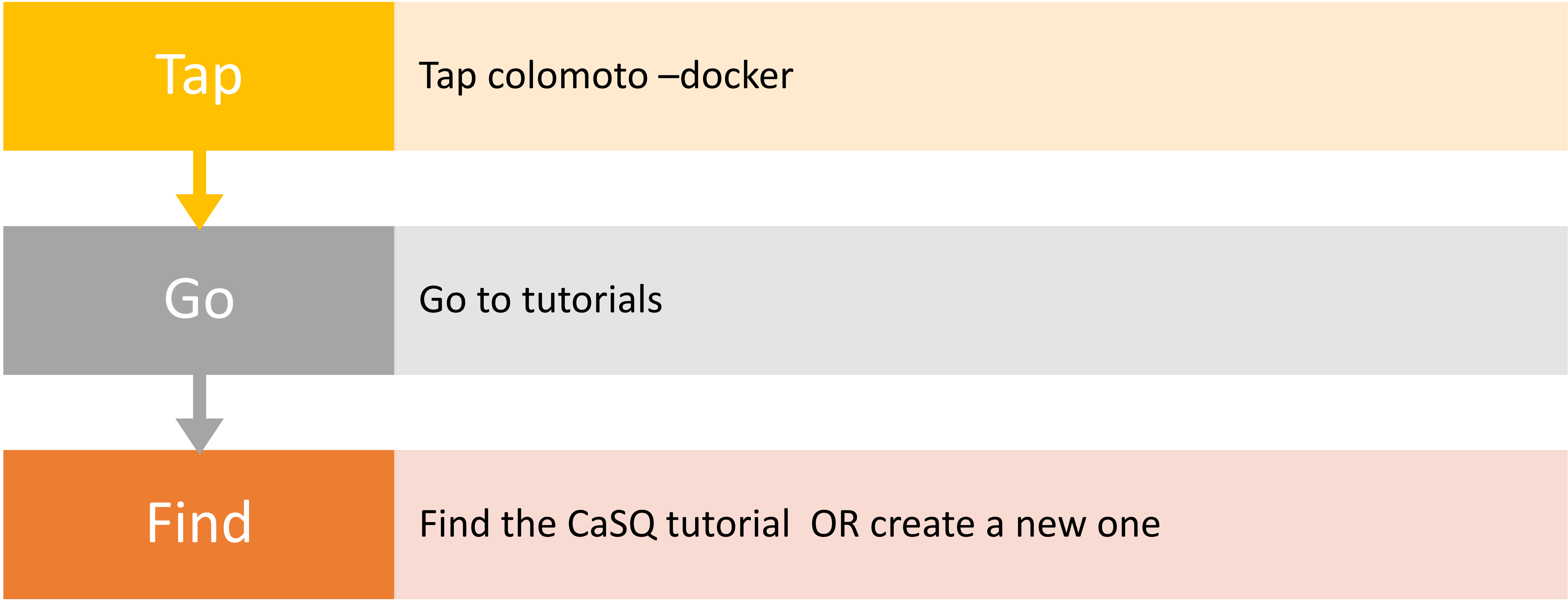


Embedded software

Besides the Jupyter notebook, the docker image provides access to the following softwares:

Software tool	Homepage	Description	Jupyter interface
ActoNet	https://github.com/algorecell/pyActoNet	Abduction-based control of fixed points of Boolean networks	Python module `actonet`
bioLQM	http://colomoto.org/biolqm/	Logical Qualitative Modelling toolkit	Python module `biolqm`
boolSim	https://www.vital-it.ch/research/software/boolSim	Attractors and reachable sets in synchronous and asynchronous Boolean networks	Python module `boolsim`
CABEAN	https://satoss.uni.lu/software/CABEAN/	A Software Tool for the Control of Asynchronous Boolean Networks	Python module `cabean`
Caspo	https://bioasp.github.io/caspo/	Reasoning on the response of logical signaling networks with Answer Set Programming	Python module `caspo_control`
CaSQ	https://github.com/soli/casq	Convert static interaction maps into executable models	Python module `casq`
CellCollective	https://cellcollective.org	Model repository and knowledge base	Python module `cellcollective`
GINsim	http://ginsim.org	Boolean and multi-valued network modelling	Python module `ginsim`
MaBoSS	http://maboss.curie.fr	Markovian Boolean Stochastic Simulator	Python module `maboss`
mpbn	https://github.com/pauleve/mpbn	Most Permissive Boolean Networks	Python module `mpbn`
NuSMV	http://nusmv.fbk.eu	Symbolic model-checker	Python module `nusmv`
Pint	https://loicpauleve.name/pint	Static analyzer for dynamics of Automata Networks	Python module `pypint`
PyBoolNet	https://github.com/hklarner/PyBoolNet	Generation, modification and analysis of Boolean networks	Python module `PyBoolNet`
R-BoolNet	https://cran.r-project.org/package=BoolNet	Analysis and reconstruction of Boolean networks dynamics	RPY2 python interface
StableMotifs	https://github.com/jgtz/StableMotifs	Target-control of Boolean networks	Python module `stablemotifs`

Open your shell



Tap

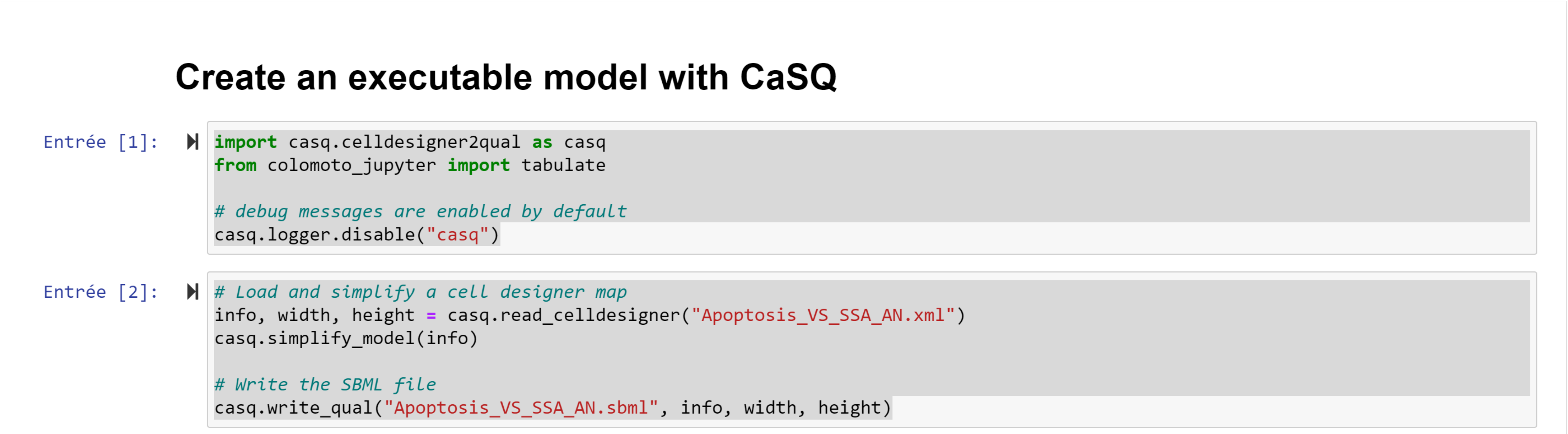
Tap colomoto –docker

Go

Go to tutorials

Find

Find the CaSQ tutorial OR create a new one



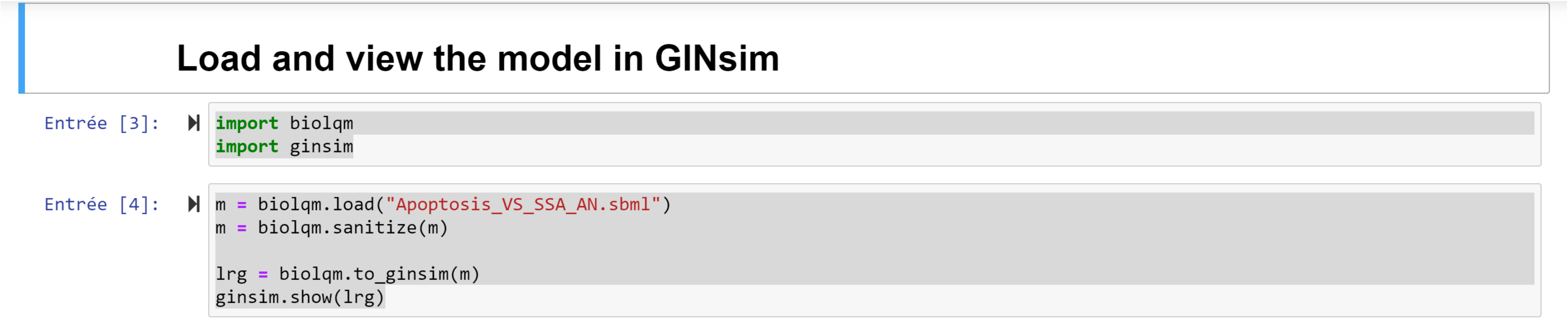
Create an executable model with CaSQ

Entrée [1]:

```python
import casq.celldesigner2qual as casq
from colomoto_jupyter import tabulate

# debug messages are enabled by default
casq.logger.disable("casq")
```

Entrée [2]:

```python
# Load and simplify a cell designer map
info, width, height = casq.read_celldesigner("Apoptosis_VS_SSA_AN.xml")
casq.simplify_model(info)

# Write the SBML file
casq.write_qual("Apoptosis_VS_SSA_AN.sbml", info, width, height)
```



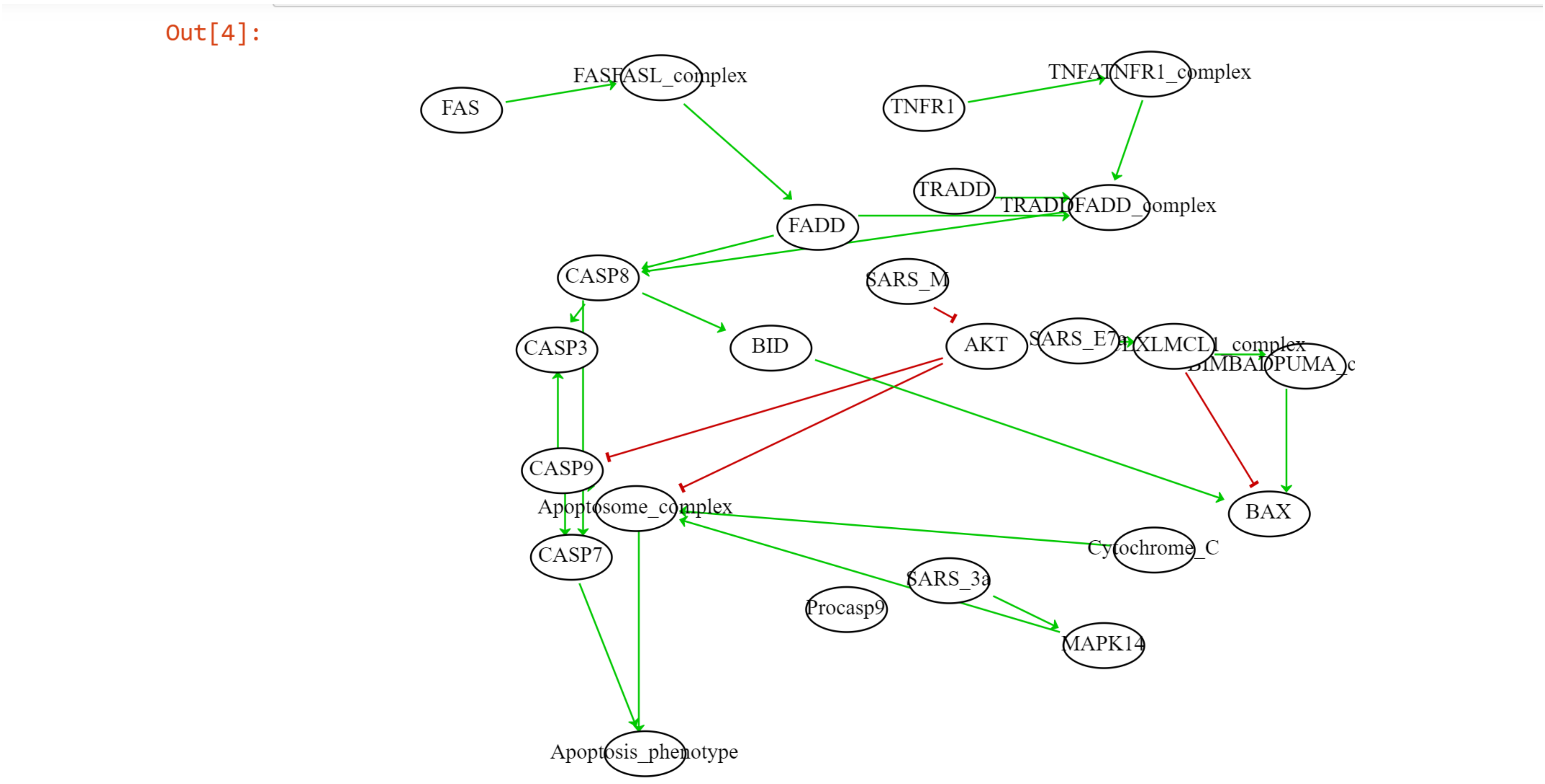
Load and view the model in GINsim

Entrée [3]:

```python
import biolqm
import ginsim
```

Entrée [4]:

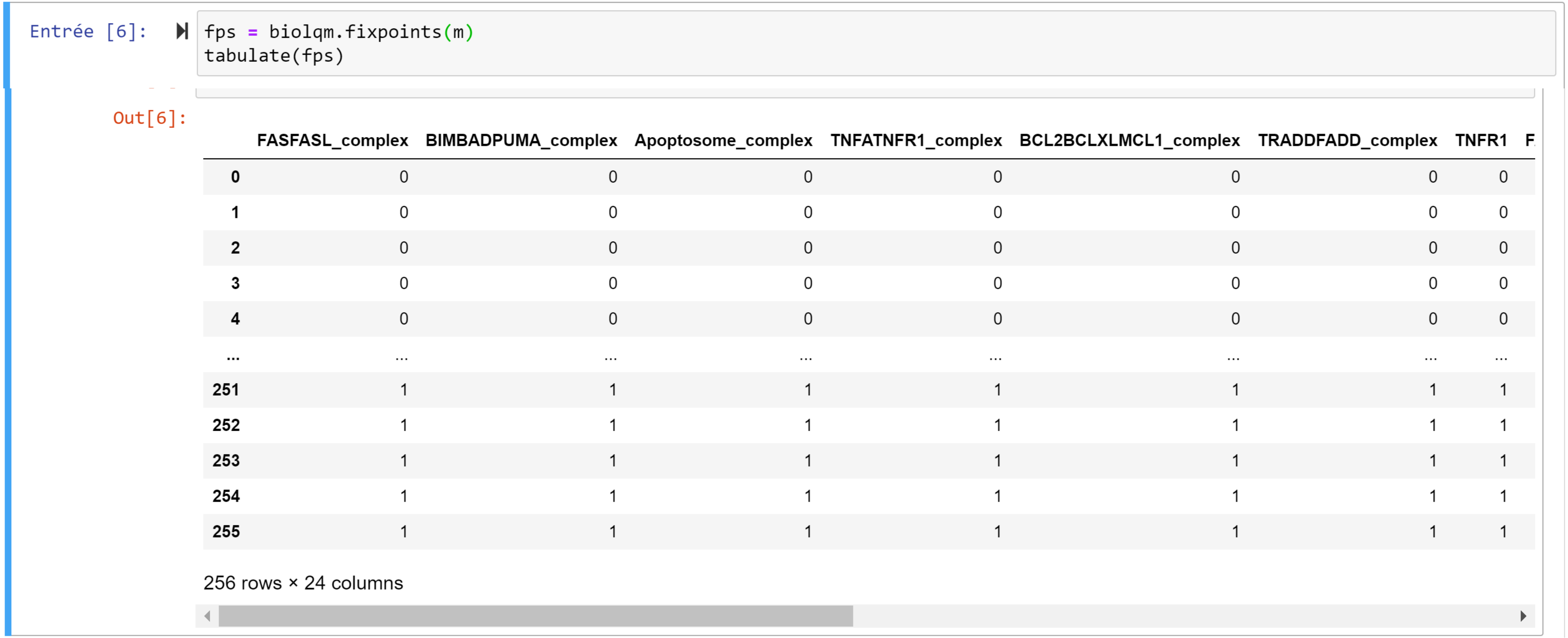
```python
m = biolqm.load("Apoptosis_VS_SSA_AN.sbml")
m = biolqm.sanitize(m)

lrg = biolqm.to_ginsim(m)
ginsim.show(lrg)
```



Out[4]:

FAS
FASFASL_complex
TNFR1
TNFATNFR1_complex
TRADD
TRADDFADD_complex
FADD
CASP8
SARS_M
CASP3
BID
AKT
SARS_E7a
XLMCL1_complex
BIMBADPUMA_c
CASP9
Apoptosome_complex
BAX
Cytochrome_C
CASP7
SARS_3a
Procasp9
MAPK14
Apoptosis_phenotype

Calculate stable states



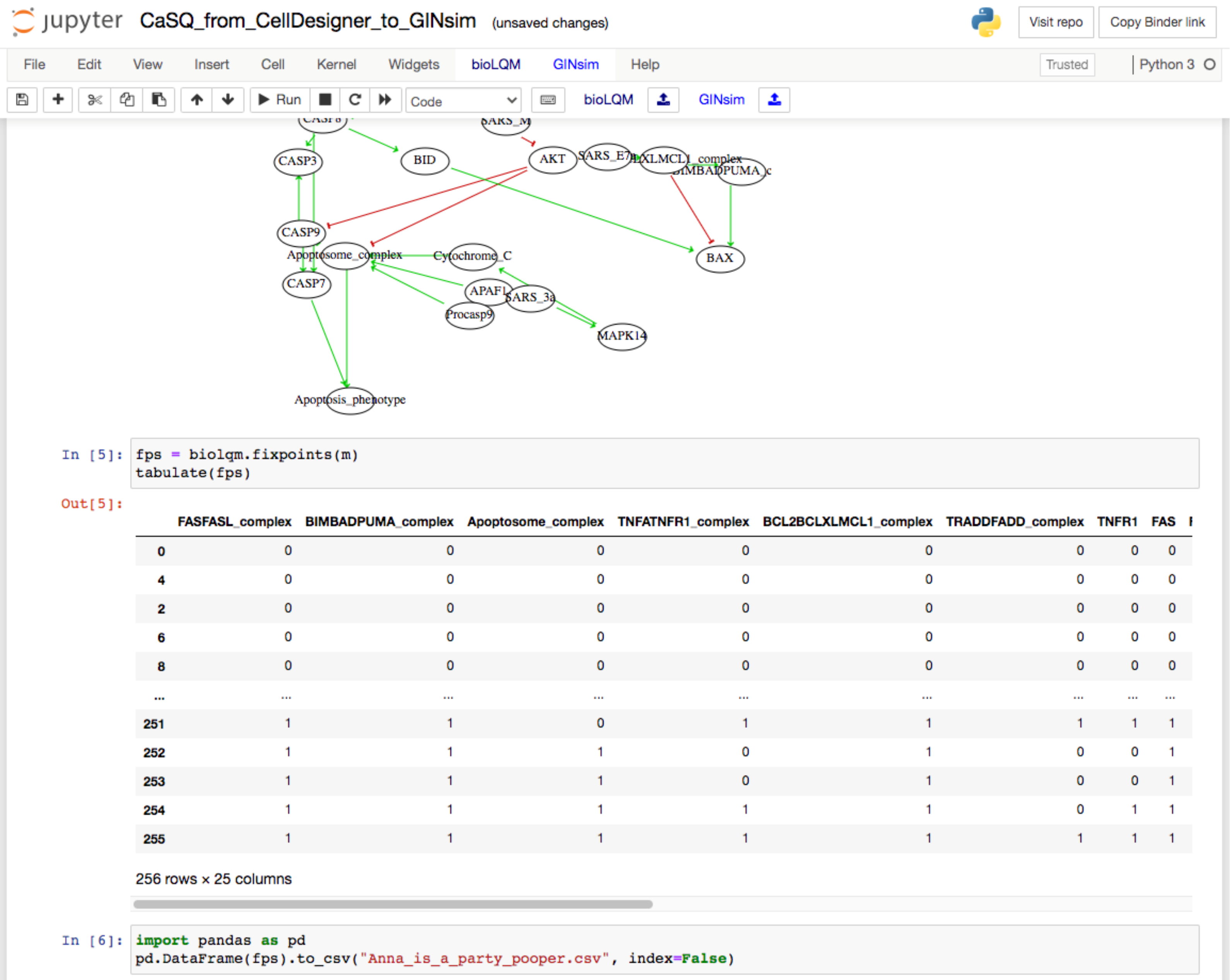
Entrée [6]:

```
fps = biolqm.fixpoints(m)
tabulate(fps)
```

Out[6]:

	FASFASL_complex	BIMBADPUMA_complex	Apoptosome_complex	TNFATNFR1_complex	BCL2BCLXLMCL1_complex	TRADDFADD_complex	TNFR1	F
0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	
...	...	...	...	...	...	...	...	
251	1	1	1	1	1	1	1	
252	1	1	1	1	1	1	1	
253	1	1	1	1	1	1	1	
254	1	1	1	1	1	1	1	
255	1	1	1	1	1	1	1	

256 rows × 24 columns



```
In [5]: fps = biolqm.fixpoints(m)
        tabulate(fps)
```

Out[5]:

	FASFASL_complex	BIMBADPUMA_complex	Apoptosome_complex	TNFATNFR1_complex	BCL2BCLXLMCL1_complex	TRADDFADD_complex	TNFR1	FAS
0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...
251	1	1	0	1	1	1	1	1
252	1	1	1	0	1	0	0	1
253	1	1	1	0	1	0	0	1
254	1	1	1	1	1	0	1	1
255	1	1	1	1	1	1	1	1

256 rows × 25 columns

```
In [6]: import pandas as pd
        pd.DataFrame(fps).to_csv("Anna_is_a_party_pooper.csv", index=False)
```